
ASSIGNMENT 6

To complete the assignment for today you need to download the OpenSesame file you recently uploaded for Assignment 5. You can download it via canvas.vu.nl.

Once you are finished with this assignment, please save the file as

`Asgn_6_<here your name>.osexp`

(for example `Asgn_6_vanmoorselaar.osexp`)

before submitting it to Canvas. Good luck!

Start the first inline script with:

- 1) In your Stroop task experiment, add one more value called “**disco**” to *block_type*. This is going to be a more challenging question. Instead of responding to the *color* of the word, participants are now to respond to the *identity* of the word, which will be either “RED”, “GREEN” or “BLUE”. Of this word, every letter will be presented in a different color. Basically, this means that you can’t use

```
canvas.text(word_identity, color=word_color)
```

anymore in its current form, as that only allows you to print a word in a single color. You’ll have to write a loop that iterates through the letters of the word which got picked and prints each letter in a different color. You will also need to use the *x* argument of `canvas.text()`, because each letter has to be drawn to the right of the previous one (if you omit this *x* all letters will be printed in the center on top of each other), So in *pseudo code*, the loop that you will have to create, would look like this:

```
calculate a starting value for x
for each letter in word_identity:
    pick a random color
    print the current letter using the canvas.text() function at position x
    increase x by a certain amount of pixels
```

On a more general note: It is always useful to first write down complex programming problems in *pseudo code*. You could regard this as an intermediate form between natural language and formal Python code. First describe the problem in plain English (or any other preferable language), so you get the grasp of the task and problem that needs to be done. Try to write it in such a way, that each line you write would constitute a statement in Python (this makes it easier to translate it later). Doing it this way, pseudo code can really help you organize your thoughts and give you an overview of which concrete steps you will need to take in a language you naturally understand. It is then only a small step to translate your pseudo code to real Python code. So if you can make neither heads nor tails of a programming problem in the future: start with pseudo code!

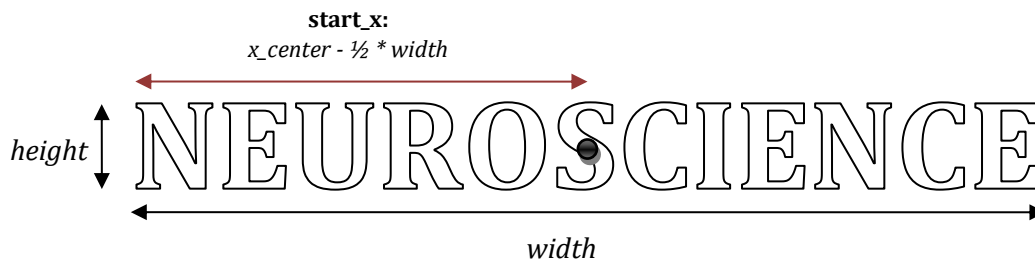
The starting value for x can be determined by a useful canvas function that calculates the size of a string of text for you. Go to <https://osdoc.cogsci.nl/3.2/manual/python/canvas/> and see if you can find this function that calculates the text size for you.

a) What is this function called?

b) There is also a function that gives you the value of the x-coordinate of the center of the screen. What is this function called?

- You only need this information if you are not using uniform coordinates. Otherwise the center of the screen is at (0,0)

You can use the function of a) to determine how high and wide a string is in *pixels*. The function will return this data as a *tuple* of which the first item is the *width* of the string, and the second item is the *height* of the string. We only need the first item (as the string will be automatically centered vertically (or *centered on the y-plane* as they sometimes call it)). The following figure shows how these values for instance would correspond to a string “NEUROSCIENCE”:

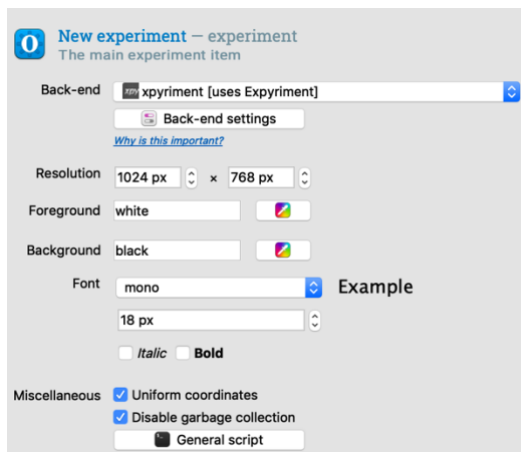


As you can see, the x coordinate that marks that start of a string (or the left-most pixel of the left-most letter of a string) is the center x coordinate of the string *minus* half its width.

c) With help of the figure above and considering that the center of the string (the big black dot) has to be at the center of the screen, how would you calculate the starting value of x for a string with help of the function from a)?

d) Knowing the width of the string and the number of letters that it contains, can you also calculate the amount of pixels you have to increase x with each cycle?

Note that in the general properties that you must enable the **uniform coordinates** box, so that the center of the screen refers to (0,0) [i.e., x,y]:



e) You now should have all the pieces of information to write the *pseudo code* I specified earlier in real Python code, and draw a string of which each letter has a random color to the canvas.

f) In addition to the above, there are three other important parts of the experiment you need to adapt for making the new disco-block function correctly:

1. Add an instruction slide before each block to tell participants to respond to the *color* of the word or to the identity of the word (depending on the *block_type* of course). To do this, add a new sketchpad to your *Block_sequence*. Then, in your *Block_list*, you can add a new variable called *block_instruction*. Here you specify the instruction for the participants. Then, when you add text to your sketchpad, you can get this information by using `[block_instruction]`.
2. The *correct_response* variable is now set according to the *color* of the word. When the *block_type* is *disco*, *correct_response* should be also be set according to the identity of the word. Add an if-statement for when *block_type* is *disco* and adapt the *correct_response* variable accordingly.
3. The default `canvas.text(word_identity, color=word_color)` does not work correctly anymore for a disco block. Therefore, you'll need to check whether block type is *disco* and add a statement that changes this procedure for *disco* accordingly (it should stay the same for every other block).

g) Run your experiment and see if your code works and all blocks appear correctly, with the correct instruction slides, and the correct response registration. You can use: `print var.correct` to print the correct response. Also provide feedback about the participants' performance at the end of each block, and tell them that they can have a break.

BONUS QUESTION

h) For this bonus question I am assuming that you have written the code in response to question **e** such that the color of each letter was selected random. The downside of this is that simply by chance, although unlikely, all letters will have the same color. Adjust your code such that adjacent letters by design have different colors. If you have already taken this into account when answering question **e** than today there is no bonus question 😊